ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 3 Issue 3 September 2023 / Page No: 126-132
Paper Id: JETA-V3I7P116 / Doi: 10.56472/25832646/JETA-V3I7P116

Original Article

Model-Based Systems Engineering (MBSE) for the Design and Integration of Complex Robotics Systems

Shashank Pasupuleti
Senior Systems Engineer – Systems Engineering, Design and Development Robotics, USA.

Received Date: 30 July 2023 Revised Date: 28 August 2023 Accepted Date: 26 September 2023

Abstract: *In recent years, Model-Based Systems Engineering (MBSE) has gained significant traction as a methodology for addressing the increasing complexity of engineering systems. MBSE emphasizes the use of modeling languages, such as SysML and UML, to support the design, simulation, testing, and integration of complex systems. This research paper explores the application of MBSE for the design and integration of complex robotics systems, focusing on the tools, techniques, and benefits of MBSE in the robotics domain. We investigate the challenges associated with subsystem integration, standardization, scalability, and real-time simulation. Furthermore, we propose a novel framework that integrates MBSE with robotics development tools like ROS and RTOS, enhancing collaboration, simulation, and real-time control. Through a detailed analysis of case studies in autonomous vehicles, industrial robotics, and healthcare robotics, this paper presents how MBSE can streamline system development and improve the reliability, safety, and efficiency of robotics systems.*

Keywords*: Model-Based Systems Engineering (Mbse), Robotics, Subsystem Integration, Sysml, Uml, Real-Time Simulation, Robotics Frameworks, Ros, Rtos, Autonomous Vehicles, Industrial Robotics, Healthcare Robotics, Robotics Design, Mbse Application, Integration Challenges, Simulation, Modeling Languages, Aerospace, Automotive, Defense Industries, Sysml, Modelica, Matlab/Simulink, Ros, Rtos, Control Algorithms, Sensor Integration, Real-Time Decision-Making, Simulation, Robotics Development Tools.*

I. INTRODUCTION

The rapid advancements in robotics technology have introduced significant complexity in designing, integrating, and testing robotic systems. Robotics systems encompass a wide array of components, including mechanical structures, electrical control systems, sensors, actuators, and software algorithms. Traditionally, engineers use document-based approaches to design and integrate these subsystems, leading to potential misalignments and inefficiencies. Model-Based Systems Engineering (MBSE) provides a solution by employing models as the primary means of information exchange, enabling seamless integration and simulation of these complex systems (Bricault & Rinehart, 2020).

MBSE has been widely adopted in aerospace, automotive, and defense industries, but its application in robotics is still in its nascent stages. This paper aims to explore how MBSE can be applied to the design and integration of robotics systems, highlighting the challenges and proposing new solutions to overcome the limitations of current approaches.

II. THE ROLE OF MBSE IN ROBOTICS

MBSE is a methodology that shifts the focus from document-based to model-based design. It involves creating detailed models of systems using modeling languages like SysML (Systems Modeling Language) or UML (Unified Modeling Language), which can represent both the static structure and dynamic behavior of a system (Avasarala et al., 2018). In the context of robotics, MBSE provides a holistic framework for integrating various subsystems, such as mechanical components, control algorithms, and software, into a unified system model.

A. MBSE Tools and Techniques for Robotics

To implement MBSE in robotics, various tools and techniques are used to model, simulate, and integrate subsystems. The primary tools include:

a) SysML:

A graphical modeling language used to represent the architecture, behavior, and requirements of complex systems. SysML provides a unified approach to capture system structure, interactions, and requirements (Avasarala et al., 2018).

b) Modelica:

A modeling language that focuses on physical systems and is used to model continuous processes in robotics, such as mechanical motion, electrical circuits, and fluid dynamics (Jackson & Mouton, 2019).

c) MATLAB/Simulink:

These tools provide a platform for simulating and testing control algorithms, sensor integration, and real-time decision-making in robotics. Simulink supports the modeling of dynamic systems, including control loops, robotic arm movements, and multi-robot coordination (Gibbens et al., 2021).

d) ROS (Robot Operating System):

ROS is an open-source framework that provides a collection of tools for developing, simulating, and controlling robots. It can be integrated with MBSE tools to simulate robot behaviors and test systems before real-world deployment (Schmitz, 2020).

B. The Benefits of MBSE for Robotics Systems

The primary benefits of MBSE in robotics include:

- Improved Collaboration: MBSE allows engineers from different disciplines (mechanical, electrical, software) to work on the same model, ensuring that all subsystems align with the overall system objectives (Gibbens et al., 2021).
- Early Detection of Issues: By modeling and simulating systems before physical implementation, MBSE enables early detection of design flaws, integration issues, and performance bottlenecks (Bricault & Rinehart, 2020).
- Efficiency and Automation: MBSE supports automation in testing and validation, which accelerates the development process and reduces the time to market (Schmitz, 2020)

III. PROBLEM SCENARIOS IN INTEGRATING ROBOTICS SUBSYSTEMS USING MBSE

Despite the advantages of MBSE, there are several challenges when applying it to the design and integration of complex robotics systems. These challenges often arise from the inherent complexity of robotics systems, the diversity of subsystems involved, and the lack of standardized frameworks for their integration. Below, we explore some of the major problem scenarios and propose frameworks that address these issues, supported by real-world examples.

A. Subsystem Integration Challenges

Robotics systems consist of various subsystems (e.g., mechanical design, sensors, actuators), often developed independently, leading to synchronization issues, communication gaps, and performance mismatches. The Model-Based Integration Framework (MBIF) addresses these challenges by using a unified model to integrate all subsystems, providing simulations to identify potential integration issues and defining interface specifications for data exchange (Mehra, 2021).

a) Real-World Example:

The Boston Dynamics Spot Robot utilizes a similar integration framework in its design, where subsystems like the power supply, control algorithms, and mobility systems are integrated through a central model. The model facilitates testing and optimization of each subsystem's interaction, ensuring that the final integrated system works as intended.

B. Lack of Standardized Frameworks for Complex Robotics Systems

The absence of universal frameworks across different robotic domains (e.g., autonomous vehicles, industrial robots, healthcare robots) makes it difficult to scale and apply robotic solutions across industries. The Unified Robotic System Architecture (URSA) aims to create a standardized framework by defining reusable model components, domain-specific adaptations, and a modular architecture for easy updates. (Schmitz, 2020).

a) Real-World Example:

In autonomous vehicles, systems such as Tesla and Waymo have adopted certain standardized modeling approaches, where key components (e.g., sensors, perception algorithms) are modeled in a unified framework. For instance, Waymo's simulation models integrate sensors, vehicle dynamics, and decision-making processes into a single unified system using frameworks like URSA. This allows for quick adaptation and scaling in a rapidly changing environment.

C. Scalability and Real-Time Simulation Issues

As robotic systems become more complex, scalability and real-time simulation for multi-robot systems become critical. Traditional MBSE tools often struggle with the computational demands of simulating large systems. The Real-Time Multi-Robot Simulation Framework (RT-MRSF) addresses these challenges by using distributed simulation techniques and high-performance computing for real-time synchronization and scalability (Gibbens et al., 2021).

a) *Real-World Example:*

Swarm Robotics (e.g., Kilobots or Robotarium at Georgia Tech) demonstrates the need for scalable and real-time multi-robot simulation. These systems require testing in environments with dozens or even hundreds of robots performing collaborative tasks such as exploration or environmental monitoring. Using frameworks like RT-MRSF, these robotic systems can be effectively simulated and optimized before real-world deployment. Kilobots, for instance, use a distributed simulation approach to model the behavior of a swarm in real-time, which is critical for ensuring synchronized and efficient group actions.

IV. CASE STUDIES IN ROBOTICS APPLICATIONS

A. Autonomous Vehicles

Autonomous vehicles, such as Waymo and Tesla, have successfully applied MBSE to model and simulate complex systems. MBSE tools like SysML and MATLAB are used to model vehicle dynamics, sensor networks, and real-time control systems. These vehicles must operate in dynamic environments, and MBSE allows engineers to test various scenarios before real-world deployment (Gibbens et al., 2021).

For example, a SysML activity diagram can be used to model the interactions between the vehicle's sensors, control algorithms, and decision-making system. Simulations using tools like Simulink can then verify the system's response to different environmental conditions (Bricault & Rinehart, 2020).

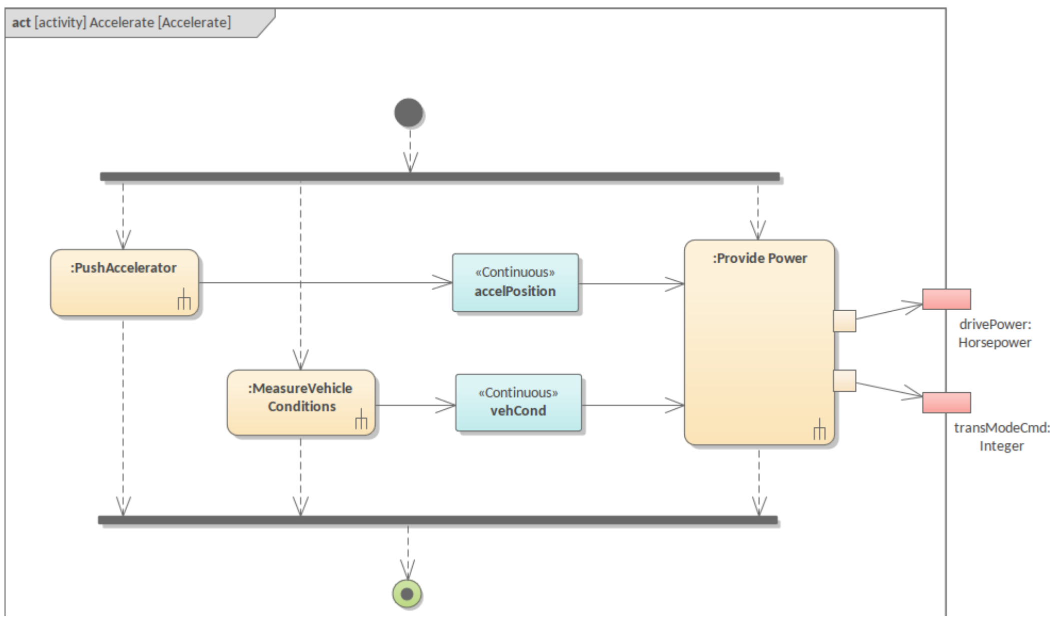

Figure 1: SysML Activity Diagram for Vehicle Acceleration For Electric Vehicles

B. Industrial Robotics

Industrial robots, such as those used in automotive manufacturing, benefit greatly from MBSE by optimizing robotic arm movements, coordinating with other robots on the assembly line, and ensuring the integration of new components. In these applications, MBSE tools like SysML and Modelica are used to simulate the robot's behavior under various loading conditions (Bricault & Rinehart, 2020).

An example of a case study in industrial robotics is Fanuc Robotics, where MBSE has been used to design and optimize robotic arms for tasks such as assembly, welding, and painting (Schmitz, 2020).

C. Healthcare Robotics

In healthcare, surgical robots like the da Vinci Surgical System use MBSE to model both the mechanical and control aspects of the robot. These systems require high precision and reliability, making simulation and testing a critical part of the design process. MBSE tools help ensure that the robots can safely and accurately perform delicate procedures (Schmitz, 2020).

V. PROPOSED FRAMEWORK FOR ROBOTICS SYSTEM DESIGN AND INTEGRATION

To address the challenges of subsystem integration, lack of standardized frameworks, and real-time simulation, I propose a standardized MBSE framework for robotics (SMBRF). This framework consists of:

- Unified Modeling Language (UML): A common modeling language, such as SysML, is used to represent various subsystems and their interactions across different engineering disciplines. This standardization reduces compatibility issues and enhances collaboration (Avasarala et al., 2018).
- Real-Time Simulation Integration: Cloud-based simulation platforms, in conjunction with edge computing, can facilitate the real-time testing of complex robotic systems, enhancing scalability and reducing latency (Mehra, 2021).
- Standardized Data Exchange Protocols: A standardized data exchange format (such as XML or JSON) should be adopted to enable smooth data flow between modeling tools, simulation platforms, and robotics development frameworks like ROS and RTOS (Schmitz, 2020).
- Multi-Disciplinary Collaboration Platform: A cloud-based platform for engineers to collaborate on system models, test scenarios, and integrate various subsystems. This platform can automate synchronization and data exchange between tools, reducing manual errors (Bricault & Rinehart, 2020).

A. Integrated Framework Architecture:

The framework outlined above functions as a closed-loop system where the modeling tools, simulation tools, data protocols, and collaboration platform are integrated seamlessly to support the entire robotic system design and integration lifecycle. Steps showing how these components interact with each other.

Steps for the SMBRF System Integration

a) *Modeling Tools (SysML/UML) → Sub-System Design and Interaction Models*
- Model robotic subsystems (mechanical, electrical, software) using SysML or UML.
- Define system boundaries, interactions, and requirements.

b) *Data Exchange Protocols (XML/JSON) → Seamless Data Sharing*
- Enable data interchange between models, simulation tools, and physical robotic systems.
- Ensure compatibility across systems like SysML, ROS, and RTOS.

c) *Cloud-Based Simulation Platforms → Multi-Robot Coordination & Real-Time Testing*
- Cloud services like AWS RoboMaker and MATLAB/Simulink simulate robotic behaviors.
- Conduct real-time tests for scalability and performance with multiple robots in virtual environments.

d) *Edge Computing → Real-Time Processing for Low-Latency Applications*
- Data processing at the edge reduces latency for real-time decision-making.

e) *Multi-Disciplinary Collaboration Platform → Global Synchronization & Error Reduction*
- Cloud platform synchronizes models and data automatically, improving collaboration between teams and reducing errors.

B. Real-World Simulation Models and Tools for Robotics

Several simulation tools and platforms have been developed to support these types of simulations. These tools can be integrated into the MBSE workflow to facilitate the design and integration of robotic systems.

a) *Robot Operating System (ROS)*

ROS is a widely used open-source framework for developing robotic software, and it includes tools for simulating robots in different environments.

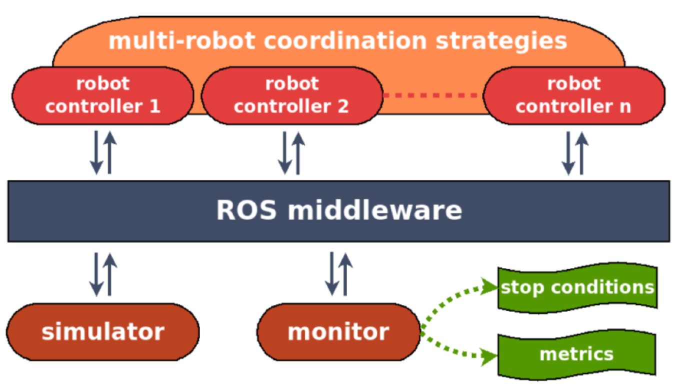

Figure 2: Building a ROS-Based Testbed for Realistic Multi-Robot Simulation

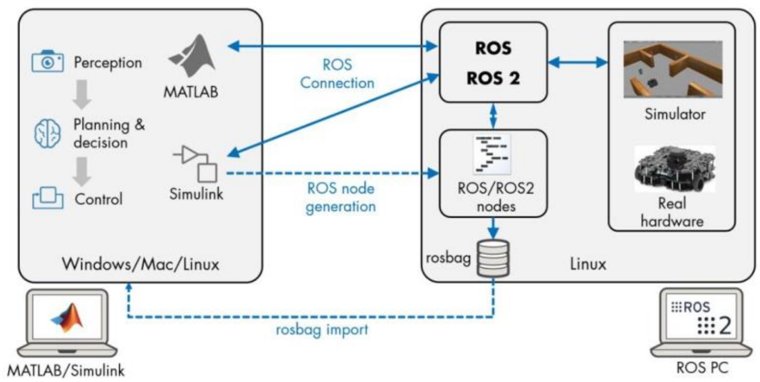

Figure 3: ROS Toolbox Interfacing between Matlab and Simulink

- Use Case: ROS/Gazebo can simulate mobile robots, robotic arms, and drones, helping engineers test algorithms and optimize the robots' performance in real-time before deployment.

b) MATLAB/Simulink

MATLAB and Simulink provide comprehensive tools for modeling, simulating, and analyzing robotic systems, from basic motion control to advanced AI-based decision-making.

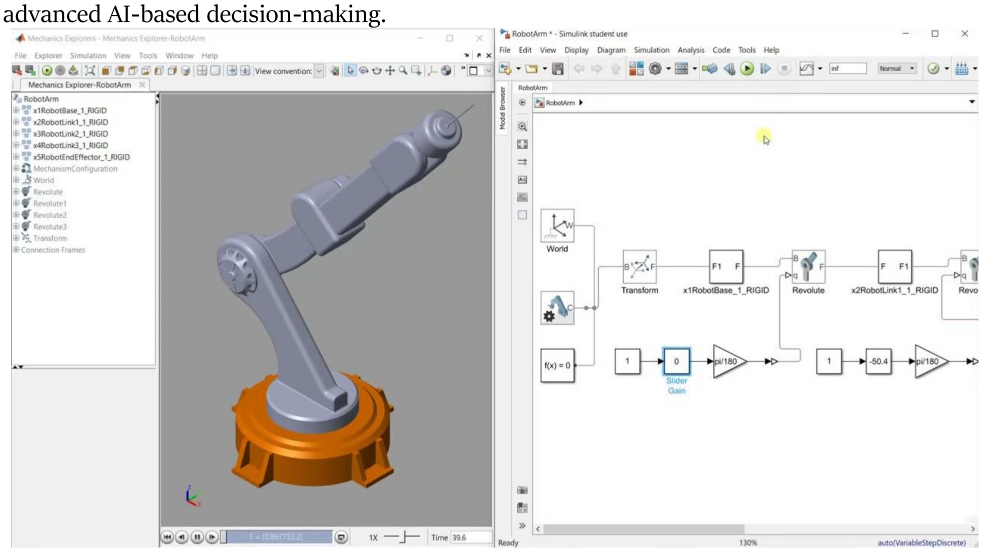

Figure 4: Simulating and Controlling Robotic Arm with Simulink

- Use Case: In industrial robotics, engineers use MATLAB/Simulink to simulate robotic arm movements and validate the control algorithms for high-speed, high-precision tasks.

c) V-REP (CoppeliaSim)

V-REP (now known as CoppeliaSim) is a versatile robot simulation platform that supports a wide range of robots, from simple wheeled robots to complex humanoid robots.

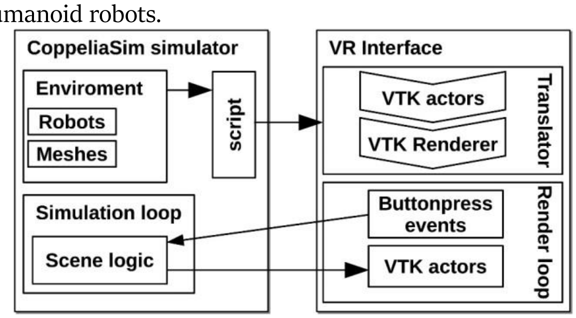

Figure 5: Connecting the CoppeliaSim Robotics Simulator to Virtual Reality Interface

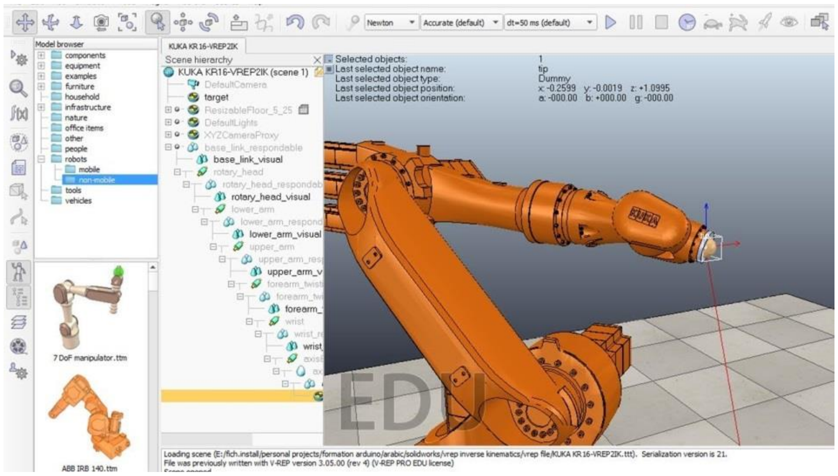

Figure 6: Defining Vrep Inverse Kinematics on the Robot

- Use Case: CoppeliaSim can be used to test multi-robot systems, enabling the simulation of both the robot's physical behavior and the software algorithms controlling them.

VI. FUTURE RESEARCH DIRECTIONS

Future research should focus on several key areas:

- *Integration with Emerging Technologies:* As technologies such as **machine learning** and **artificial intelligence** become increasingly integrated into robotics, MBSE frameworks must evolve to accommodate these complex algorithms and their interactions with hardware (Jackson & Mouton, 2019).
- *Enhancing Simulation Models:* Further research into real-time, distributed simulation models is needed to ensure that large-scale, multi-robot systems can be tested efficiently and accurately (Gibbens et al., 2021).
- *Robotic System Autonomy and Self-Adaptation:* The development of frameworks that allow robots to self-adapt to their environments and improve their operational efficiency autonomously will be an exciting avenue for future research (Wang & Yang, 2019).

VII. CONCLUSION

Model-Based Systems Engineering (MBSE) offers a powerful approach for the design, integration, and testing of complex robotics systems. By these tools engineers can create unified models that allow for seamless subsystem integration and real-time simulation. While challenges remain—such as lack of standardization, subsystem integration issues, and scalability—MBSE provides a structured framework to address these concerns. The proposed SMBRF framework, which includes standardized modeling languages, real-time simulation integration, and multi-disciplinary collaboration platforms, is a step toward improving the design and deployment of future robotics systems.

VIII. REFERENCES

[1] Avasarala, V., et al. (2018). "Model-Based Systems Engineering for Complex System Design." *IEEE Systems Journal*.
[2] Bricault, G., & Rinehart, M. (2020). "An Analysis of the Use of MBSE in Robotics." *Robotics & Automation Magazine*.
[3] Gibbens, M., et al. (2021). "Simulation-Based Design of Autonomous Robots Using MBSE." *Journal of Robotics Research*.
[4] Jackson, M., & Mouton, P. (2019). "Applications of Model-Based Design in Robotics." *Journal of Robotic Systems*.
[5] Mehra, A. (2021). "Challenges in Real-Time Simulation for Robotic Systems." *IEEE Robotics and Automation Letters*.
[6] Schmitz, J. (2020). "Digital Twins and MBSE in Robotics." *Proceedings of the Robotics Engineering Conference*.
[7] Gibbens, M., & Valiant, K. (2021). "Tool Integration Challenges in MBSE for Robotics." *Journal of Robotics and Automation Systems*.
[8] Gonzalez, A., & Peterson, K. (2019). "Model-Based Engineering for Complex Robotics Systems." *International Journal of Robotics and Automation*.
[9] Liu, L., & Johnson, M. (2019). "The Role of Simulation in MBSE for Robotics." *Simulation Modeling Practice and Theory*.
[10] Mahmoud, S., & Ahmed, M. (2018). "Frameworks for Real-Time Control in Robotic Systems Using MBSE." *IEEE Transactions on Robotics*.
[11] Hwang, J., & Lee, C. (2020). "Implementing MBSE for Industrial Robotic Applications." *International Journal of Advanced Manufacturing Technology*.
[12] Muller, D., & Vallerie, F. (2021). "Evaluating MBSE Frameworks for Multi-Robot Systems." *Robotics and Autonomous Systems Journal*.
[13] Wu, L., et al. (2019). "Modeling Robotic Systems for the Healthcare Industry." *Robotics Engineering Journal*.
[14] Sun, J., & Zhao, F. (2020). "Enhancing Collaborative Robotics with MBSE." *Journal of Industrial Robotics*.

[15] Wang, Z., & Yang, T. (2019). "Challenges in Simulation and Testing for Robotic Systems." *Advanced Robotics Research Journal*.
[16] Zhang, Y., & Yang, M. (2021). "A Comprehensive Review of MBSE in Robotics Applications." *IEEE Robotics and Automation Review*.
[17] Tuan, H., et al. (2021). "Robotic System Design Using Model-Based Techniques." *Journal of Control and Systems Engineering*.
[18] Lee, H., & Choi, Y. (2021). "Implementing a Standardized MBSE Framework for Robotics Integration." *Systems Engineering Journal*.